Readme

u6142160 Liyao Tang

The work is down based on the lab02 instructions and the assignment instruction. The default website to start the crawling is the "3310exp.hopto.org:9780/". Note that 'make run' command will only start from the default server. To start from a different website "domain:port/page", you can run the program as "./crawler domain port page". For example, to start from "3310exp.hopto.org:9780/20/25.html", run the complied program as "./crawler 3310exp.hopto.org 9780 20/25.html".

The crawler defaults to use TCP connection and will switch to UDP if EAI_SERVICE is reported in 'getaddrinfo()', but this function is not tested against any server yet.

The crawler will receive all messages sent from server using a dynamic-array-like structure 'Data_Buffer', assuming the message in any single packet is smaller than BUFLEN (default to 4096) bytes. This function is designed in case that a large website arrives in multiple packets but is unfortunately not tested again.

The information of each web site is recored in a link list structure 'Domain_List' and each page will be only recorded once. Web pages are recorded only in their IP addresses to avoid duplicate, assuming that one web page's IP address can hardly change during the execution time of this simple crawler. That is, Two pages will be viewed as the same page if they lead to the same IP address during the execution of my crawler. While maintaining the list, crawler visits web pages in a DFS style.

The required results will be printed at the end and some intermediate results will be printed while crawling the web. To be more specific, the required number of page is counted as unique IP addresses ever successfully resolved by the crawler, including 50x redirected pages and 404 not-found pages. The largest page is the page with the longest content length, instead of the whole received message. The most-recently modified page is the page with largest Last-Modified time which is stored in type 'struct tm' and is converted into type 'time_t' using 'mk_time()' in comparison.

Finally, the crawler is built and tested successfully on the lab machine against the friendly test server.